

# Art analysis of API Management

This has been done in 2025.

The goal is to safeguard the *API* infrastructure.

## Some ideas on how to manage an *API*

*This article* explains the *best practices* to follow on how to manage an *API* key.

## What will be used?

Here is a list of *API* that will be used:

1. *Crossref*: This one does not require an *API* key. It just encourages us to include our contact information.

Here is their recommendation following *this website*:

**## How to use the API**

**### Best practice**

By using our REST API, you are part of our community that supports open access to scholarly metadata. We ask that you are considerate of other users and don't take actions that will make the API unstable, and hence unusable for others. You can do this by:

*Caching responses so you don't make the same requests over and over.  
Be considerate about the frequency with which you refresh your cache.  
Use the mailto parameter and specify a User-Agent header  
that identifies you and your script.  
Handle HTML response codes and monitor response times.  
Back off if the response times start to increase.*

They recommend to provide an *email* in the *mailto* parameter of the request. In fact, they will send an email and not directly block the traffic if there is an issue.

Indeed, **Crossref has of a limit of 50 requests per second, among 5 concurrent requests**. The limits are written in *x-rate-limit-limit* and *x-rate-limit-interval*.

If the limits are exceeded, a *429 response* will be sent.

Besides, it seems that the responses has a maximum size limit too. The response will contain *next-cursor* that could be used in the next request to get the next publications from the next page.

Anyway, the *cursors* expire in *5minutes*, so the user will have to reload the page after 5minutes.

2. *Semantic Scholar*: This one does not require an *API* key.

The limit is hard to tell because various papers have been written and the official doc merely does not provide the actual request rate limit.

It seems that **Semantic Scholar has a limit of 100 requests every 5 minutes, per IP address.**

So I assume that it is possible to bypass the *API* rate providing different IP address.

It seems also a good idea not to spam the *API* with too many requests at the same time. **A time interval of 3ms** between each request has been introduced in **2021** by *this article*.

There is also an equivalent of the cursors for this *API*. It is just called **limit**, and the limit can be set by the one who does the request. Besides, *Semantic Scholar* also introduces the parameter called **offset**. This is well explained by *this article* from 2023. Anyway, the maximum number of results to return called *limit* must not exceed 10000, providing *offset* is not set.

A chinese researcher from the *Institute of International Rivers and Ecoscurity* from *Yunnan University (China)* just wrote a so called **Semantic Scholar miner**. *This github repository* contains the code with some explanations on the *API* in the `man` directory.

3. *TheBrain API*: This one requires an *API* key.

This one is very scary because it's like nobody ever used this *api* before, even if it seems to be well documented. In fact, I can't find any *API* rate limit. All I found is *the official release article*. Besides, here is *the github account* of *TheBrain*. I'm just scared that one day the *API* rate will be exceeded and it will ask us to pay a bill to continue using their service.

## Which side?

1. *Crossref*: used by the *user*, in the *client*, to retrieve a publication.

It is also used in the *server* to retrieve the authors behind the publication. In fact, the authors that are not from the database should not be inserted. That's why it is the *server* that should retrieve the authors, to compare with the database.

2. *Semantic Scholar* used in the *server* to get insights and statistics.

3. *TheBrain API* used in the *server* to synchronise the *R3MOB* Brain and the local database.

EOF